Xen 3.0 –

Hypervisor Technology and Hardware Support for Virtualization

Steve Hand, XenSource Inc.

GI Munich October 2006

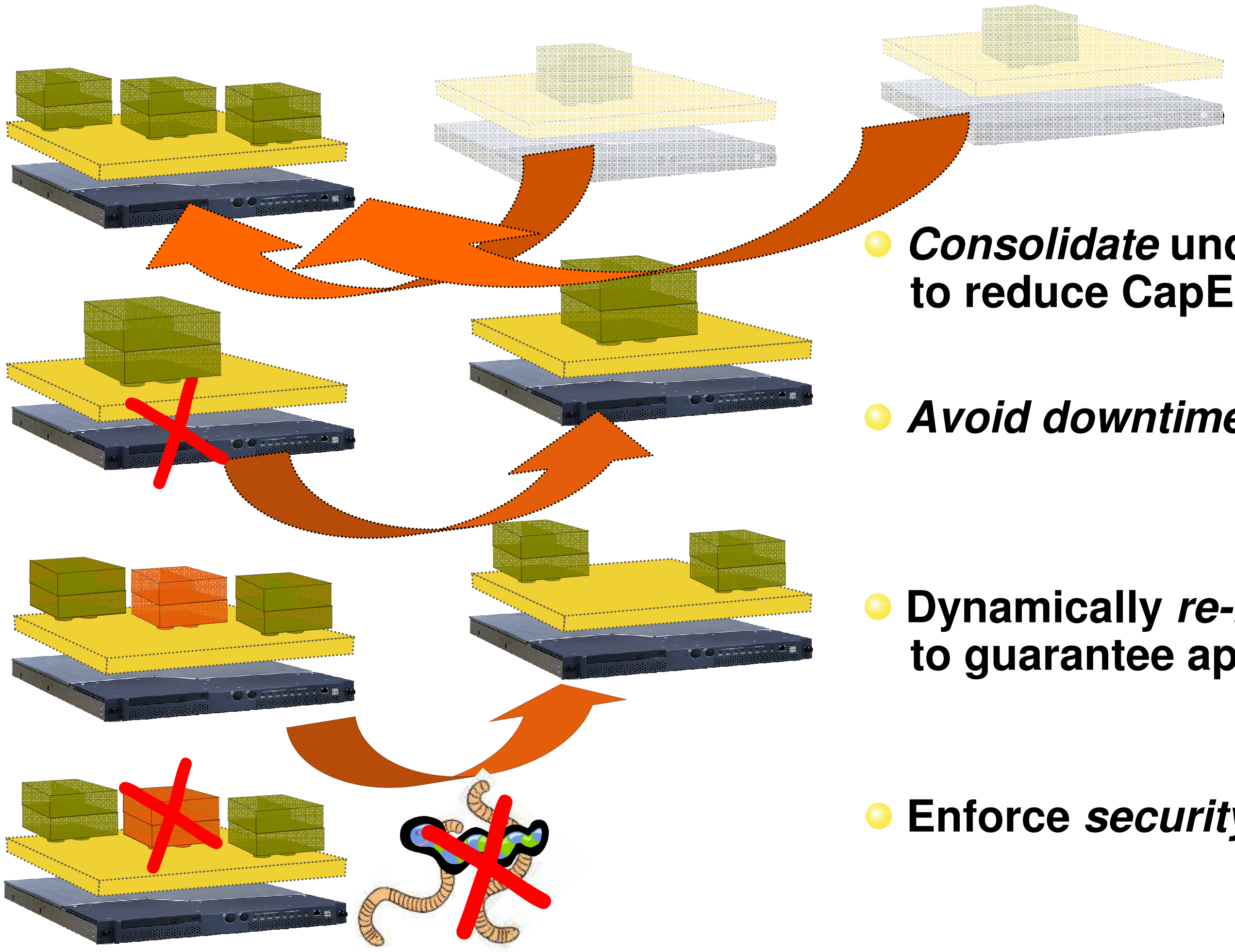
Outline

- Virtualization Overview
- Xen Architecture
- New Features in Xen 3.0
- Hardware Virtualization
- Xen Roadmap
- Questions

Virtualization Overview

- Single OS image: Virtuozo, Vservers, Zones
  - Group user processes into resource containers
  - Hard to get strong isolation
- Full virtualization: VMware, VirtualPC, QEMU
  - Run multiple unmodified guest OSes
  - Hard to efficiently virtualize x86
- Para-virtualization: UML, L4Linux, Xen
  - Run multiple guest OSes ported to special arch
  - Arch Xen/x86 is very close to normal x86

Virtualization in the Enterprise



- *Consolidate* under-utilized servers to reduce CapEx and OpEx
- *Avoid downtime* with VM Relocation
- Dynamically *re-balance workload* to guarantee application SLAs
- Enforce *security* policy

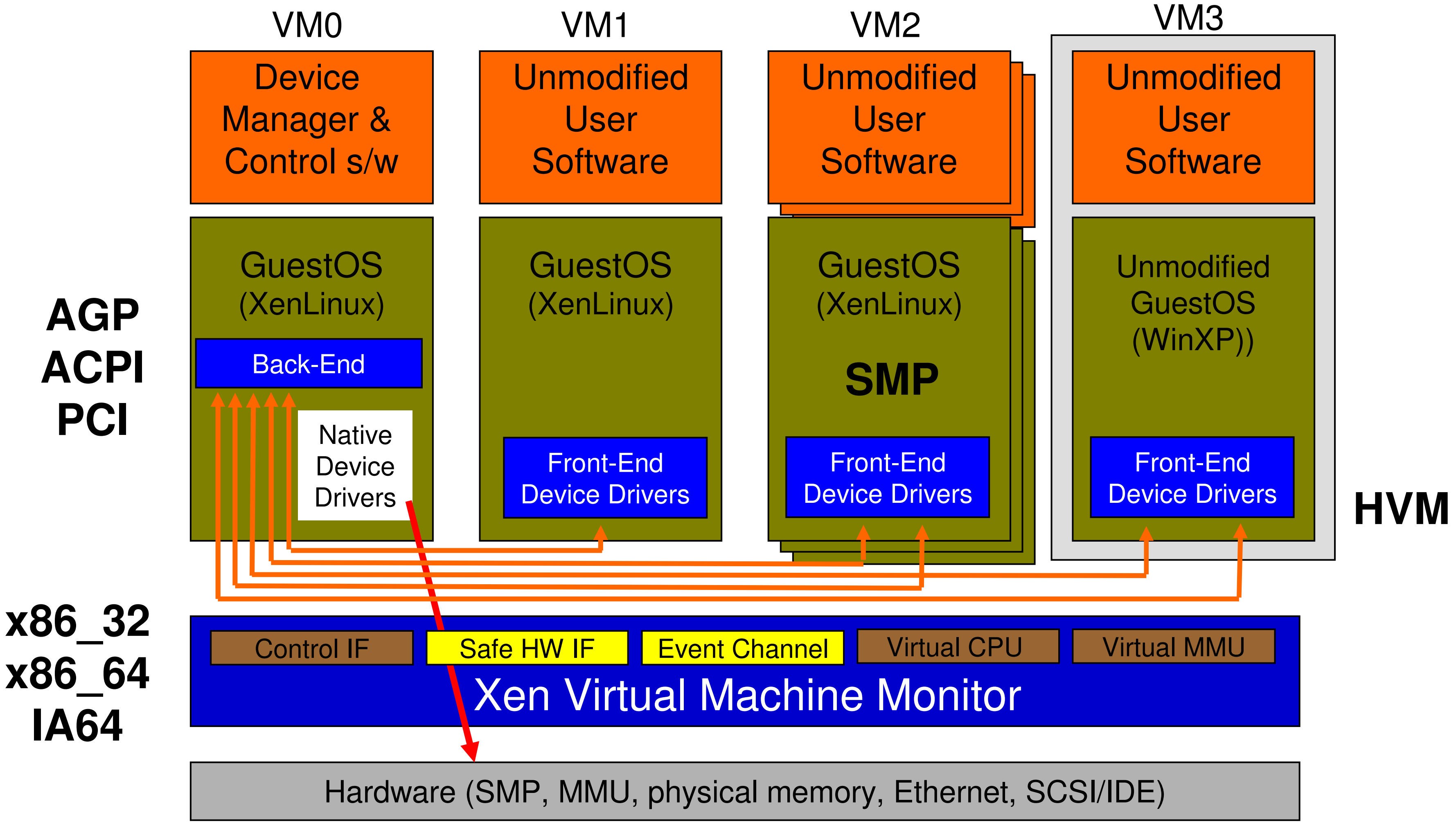
Virtualization possibilities

- Value-added functionality from outside OS:
  - Fire-walling / network IDS / “inverse firewall”
  - VPN tunnelling; LAN authentication
  - Virus, mal-ware and exploit scanning
  - OS patch-level status monitoring
  - Performance monitoring and instrumentation
  - Storage backup and snapshots
  - Local disk as just a cache for network storage
  - Carry your world on a USB stick
  - Multi-level secure systems

Xen 3.0 (5th Dec 2005)

- Secure isolation between VMs
- Resource control and QoS
- Latest stable is **3.0.3** (Oct 17th 2006)
- x86 32/PAE36/64 plus HVM; IA64, Power
- PV guest kernel needs to be ported
  - User-level apps and libraries run unmodified
- Execution performance close to native
- Broad (linux) hardware support
- Live Relocation of VMs between Xen nodes

Xen 3.0 Architecture



Para-Virtualization in Xen

- Xen extensions to x86 arch
  - Like x86, but Xen invoked for privileged ops
  - Avoids binary rewriting
  - Minimize number of privilege transitions into Xen
  - Modifications relatively simple and self-contained
- Modify kernel to understand virtualised env.
  - Wall-clock time vs. virtual processor time
    - Desire both types of alarm timer
  - Expose real resource availability
    - Enables OS to optimise its own behaviour

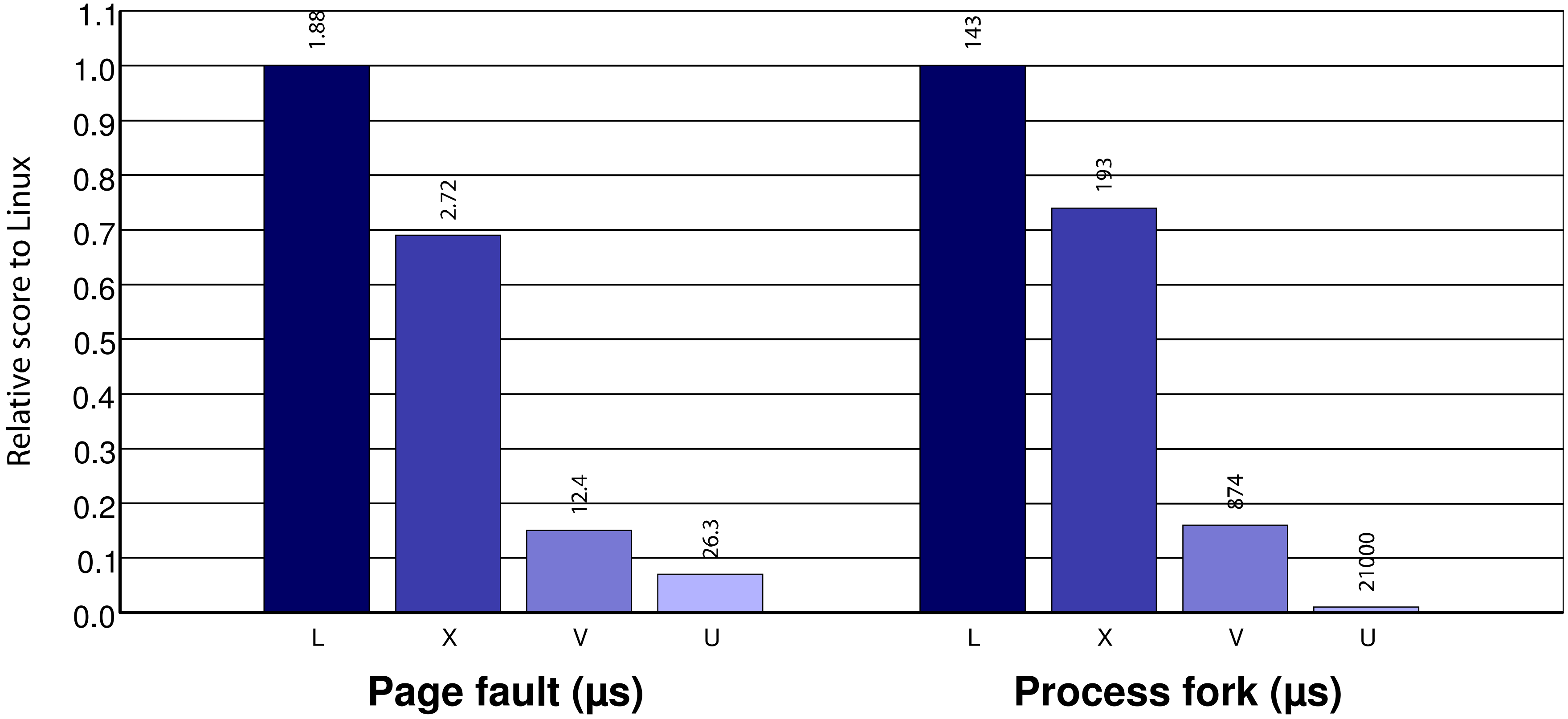
x86 CPU virtualization

- Xen runs in ring 0 (most privileged)
- Ring 1/2 for guest OS, 3 for user-space
    - GPF if guest attempts to use privileged instr
- Xen lives in top 64MB of linear addr space
    - Segmentation used to protect Xen as switching page tables too slow on standard x86
- *Hypercalls* jump to Xen in ring 0
    - Indirection via hypercall page allows flexibility
- Guest OS may install 'fast trap' handler
    - Direct user-space to guest OS system calls

Para-Virtualizing the MMU

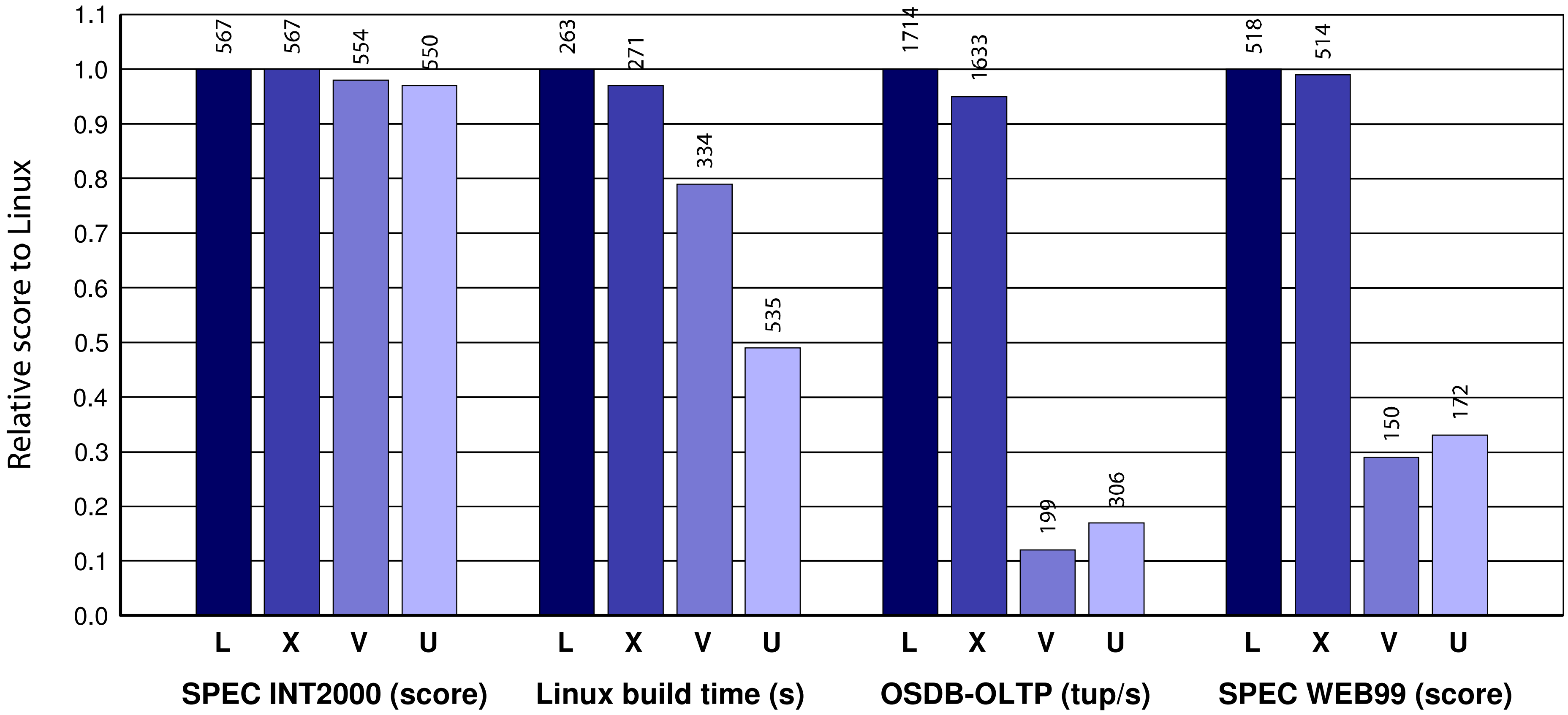
- Guest OSes allocate and manage own PTs
  - Hypercall to change PT base
- Xen must validate PT updates before use
  - Allows incremental updates, avoids revalidation
- Validation rules applied to each PTE:
  1. Guest may only map pages it owns*
  2. Pagetable pages may only be mapped RO
- Xen traps PTE updates and emulates, or ‘unhooks’ PTE page for bulk updates

MMU Micro-Benchmarks (old)



lmbench results on Linux (L), Xen (X), VMWare Workstation (V), and UML (U)

System Performance (old)



Benchmark suite running on Linux (L), Xen (X), VMware Workstation (V), and UML (U)

SMP Guest Kernels

- Xen extended to support multiple VCPUs
  - Virtual IPI’s sent via Xen event channels
  - Currently up to 32 VCPUs supported
- Simple hotplug/unplug of VCPUs
  - From within VM or via control tools
  - Optimize one active VCPU case by binary patching spinlocks
- NB: Many applications exhibit poor SMP scalability – often better off running multiple instances each in their own OS

Hardware Virtualization (1)

- Paravirtualization…
  - has fundamental benefits… (c/f MS Viridian)
  - but is limited to OSes with PV kernels.
- Recently seen new CPUs from Intel, AMD
  - enable safe trapping of ‘difficult’ instructions
  - provide additional privilege layers (“rings”)
  - currently shipping in most modern server, desktop and notebook systems
- Solves part of the problem, but…

Hardware Virtualization (2)

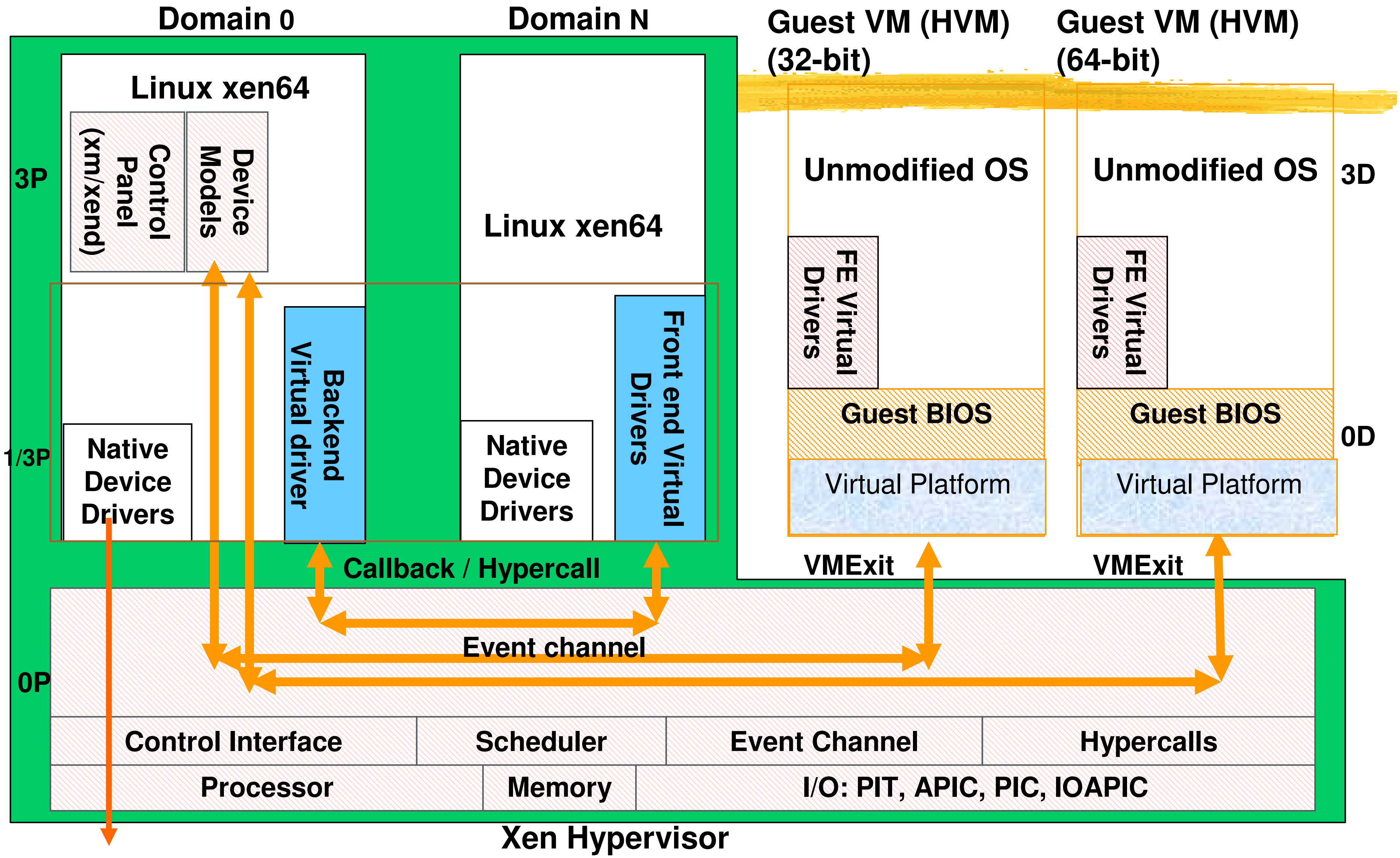
- CPU is only *part* of the system
  - also need to consider *memory* and *I/O*
- Memory:
  - OS wants *contiguous physical memory*, but Xen needs to share between many OSes
  - Need to dynamically translate between guest physical and ‘real’ physical addresses
  - Use *shadow page tables* to mirror guest OS page tables (and implicit ‘no paging’ mode)
- Xen 3.0 includes s/w shadow page tables.
- (Future x86 processors will include h/w support)

Hardware Virtualization (3)

- Finally we need to solve the I/O issue
  - non-PV OSes don’t know about Xen
  - hence run ‘standard’ PC ISA/PCI drivers
- Just *emulate* devices in software?
  - complex, fragile and non-performant…
  - … but ok as backstop mechanism.
- Better:
  - add PV (or “enlightened”) device drivers to OS
  - well-defined driver model makes this relatively easy
  - get PV performance benefits for I/O path

Xen 3: HVM

- Enable Guest OSes to be run without modification
  - E.g. legacy Linux, Solaris x86, Windows XP/2003
- CPU provides vmexits for certain privileged instrs
- Shadow page tables used to virtualize MMU
- Xen provides simple platform emulation
  - BIOS, apic, iopaic, rtc, net (pcnet32), IDE emulation
- Install paravirtualized drivers after booting for high-performance IO
- Possibility for CPU and memory paravirtualization
  - Non-invasive hypervisor hints from OS



Domain 0

Domain N

Guest VM (HVM) (32-bit)

Guest VM (HVM) (64-bit)

3P

1/3P

0P

3D

0D

Linux xen64

Control Panel (xm/xend)

Device Models

Native Device Drivers

Backend Virtual driver

Linux xen64

Native Device Drivers

Front end Virtual Drivers

Unmodified OS

FE Virtual Drivers

Guest BIOS

Virtual Platform

Unmodified OS

FE Virtual Drivers

Guest BIOS

Virtual Platform

Callback / Hypercall

VMExit

VMExit

Event channel

Control Interface

Scheduler

Event Channel

Hypercalls

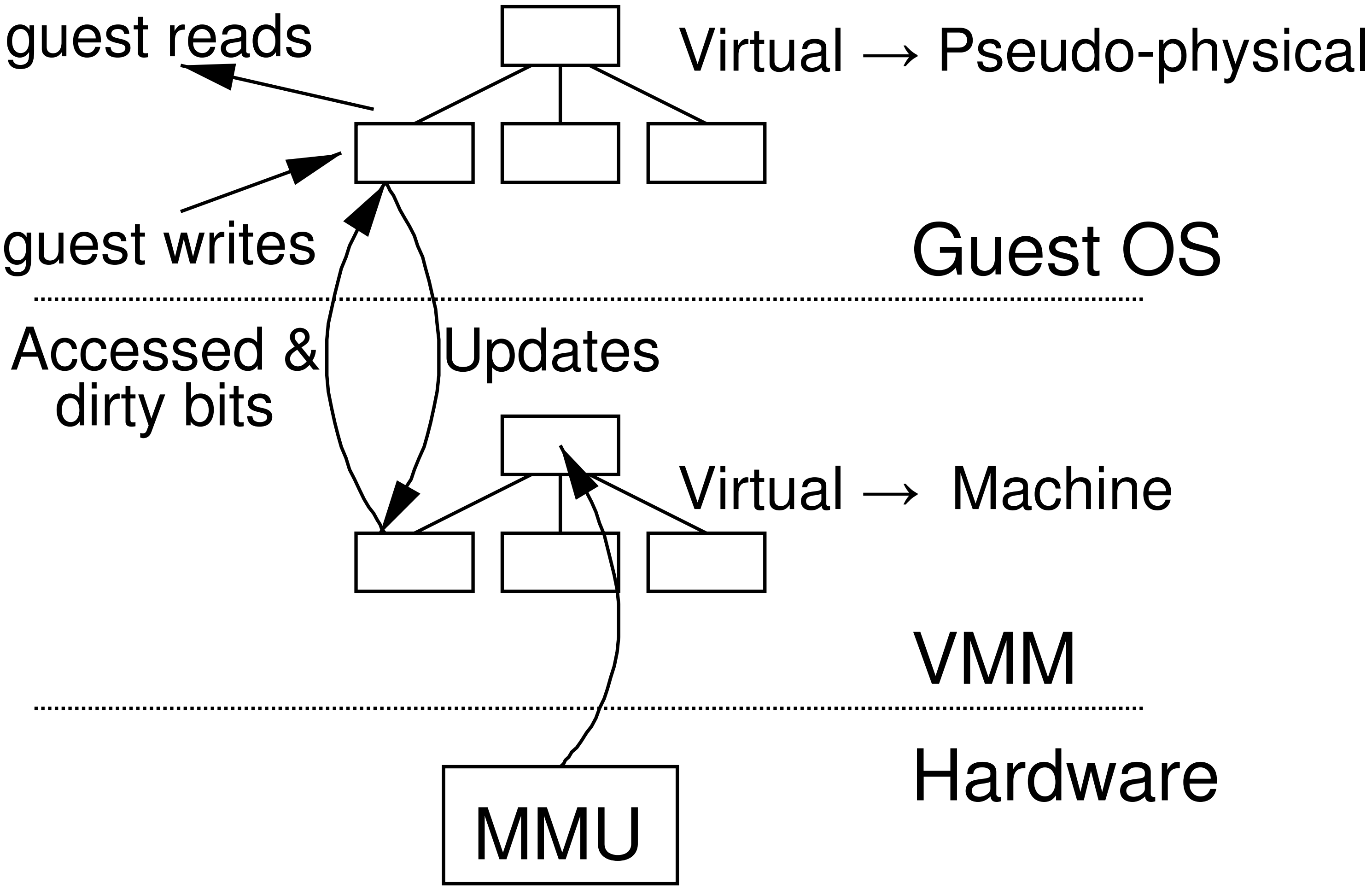
Processor

Memory

I/O: PIT, APIC, PIC, IOAPIC

Xen Hypervisor

MMU Virtualizion : Shadow-Mode



Smart I/O Hardware

- Xen 3 PV and HVM guests work with high-performance, but still a cost
  - backend s/w needed for secure multiplexing
  - can stress certain workloads (e.g. MPI)
- Next step: smart I/O for virtualization
  - make *platform* aware of virtualization
  - (e.g. additional h/w protection for DMA coming soon from Intel and AMD)
- Or make *devices* aware of virtualization…

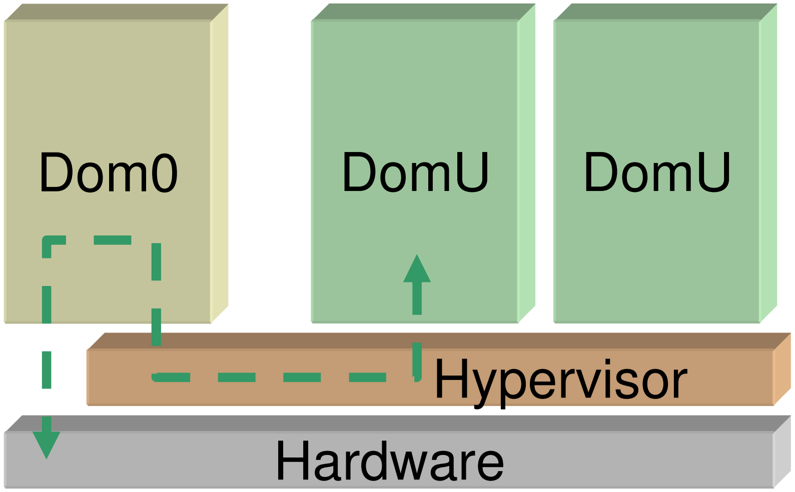
Eg: SolarFlare Solarstorm

- Solarstorm inspired by user-level networking
  - TCP/IP stack linked with user app
- Smart NIC allows safe access from guest
  - Onboard IOMMU for safe DMA
  - NIC's filter-table demuxes incoming packets to queue
  - Queues get mapped into guests
- Eliminates interrupts/syscalls/context switches
  - Can also do zero-copy tx from guests

Slides courtesy of Greg Law at SolarFlare

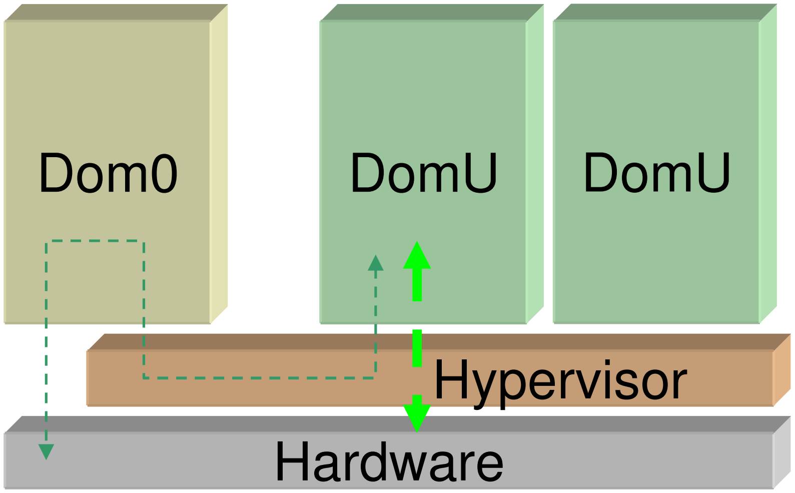
Traditional Xen: I/O via Dom0

- All 'real' drivers live in Dom0
- Guest kernels have pseudo drivers that talk to Dom0 via the hypervisor
- Necessary because only Dom0 is 'trusted'



But with SolarStorm…

- Accelerated routes set up in Dom0
- DomU can access h/w directly + safely
  - at least most of the time
  - (still slow path via Dom0)



HW Virtualization Summary

- CPU virtualization available today
  - lets Xen support legacy/proprietary OSes
- Additional platform protection imminent
  - protect Xen from IO devices
  - full IOMMU extensions coming soon
- MMU virtualization also coming soon:
  - avoid the need for s/w shadow page tables
  - should improve performance and reduce complexity
- Device virtualization arriving from various folks:
  - networking already here (ethernet, infiniband)
  - [remote] storage in the works (NPIV, VSAN)
  - graphics and other devices sure to follow…

Xen 3.x Roadmap

- Comtinued improved of full-virtualization
  - HVM (VT/AMD-V) optimizations
  - DMA protection of Xen, dom0
- Off-box management API + tools
- Performance tuning and optimization
  - Less reliance on manual configuration
- Better NUMA, Virtual framebuffer, etc
- Smart I/O enhancements

Research Roadmap

- Whole-system debugging
  - Lightweight checkpointing and replay
  - Cluster/distributed system debugging
- Software implemented h/w fault tolerance
  - Exploit deterministic replay
- VM forking
  - Lightweight service replication, isolation
- Secure virtualization
  - Multi-level secure Xen

Xen Supporters

Novell. Sun microsystems redhat. VERITAS™

Operating System and Systems Management

hp® invent IBM®

Hardware Systems

TOPSPIN®
Acquired by
CISCO SYSTEMS

intel®

AMD

Platforms & I/O

* Logos are registered trademarks of their owners

Conclusions

- Xen is a complete and robust GPL VMM
- Outstanding performance and scalability
- Excellent resource control and protection
- Vibrant development community
- Strong vendor support

Xen Source

- http://xensource.com/community

Thanks!

- Download Xen from Xen Source

http://www.xensource.com

- New stable release – Xen 3.0.3 – out now!
  - enhanced hvm support among other things.
- XenEnterprise with HVM due later this year